

GENERATIVE ARTIFICIAL INTELLIGENCE AND THE PROGRAMMED LEARNING METHOD: ENHANCING SELF-LEARNING CAPACITY IN PROGRAMMING TOWARDS LIFELONG LEARNING AND SUSTAINABLE DEVELOPMENT

Ha Viet Hai, Tran Van Long
University of Education, Hue University

Abstract: *This article analyzes the potential of integrating Generative AI with the Programmed Learning (PL) method to create a personalized and effective self-learning environment for programming. Based on the theories of PL and Self-Regulated Learning (SRL), the paper proposes a conceptual AI-PL model. An initial exploratory workshop was conducted with graduate students at Mahasarakham University (Thailand) to investigate the feasibility of using AI (ChatGPT) for self-learning Python. Preliminary feedback indicates that AI can effectively fulfill the roles of organizing, providing adaptive feedback, and coordinating the learning process as envisioned by the PL method. The approach showed potential in personalizing learning pace and content, and fostering self-learning skills. The proposed AI-PL model has the potential to overcome traditional limitations of PL, promote lifelong learning competencies, and contribute to Sustainable Development Goal 4 (SDG4) by making programming education more accessible and equitable.*

Keywords: *AI, generative artificial intelligence, programmed learning, self-learning programming, ChatGPT, self-regulated learning, personalized learning, lifelong learning, sustainable development.*

1. INTRODUCTION

Learning programming is playing an increasingly important role in schools, university, and vocational training institutions. However, teaching and learning programming often face difficulties due to the abstract nature of concepts, the requirement for strict logical thinking, and the need for immediate feedback to correct errors and reinforce knowledge during the learning process. In particular, the lack of timely feedback can lead to discouragement and reduced learning motivation of learners.

In that context, the programmed learning method emerges as a classic but still relevant pedagogical solution, which once offered reasonable principles to address these challenges. PL emphasizes personalizing the learning process, breaking down knowledge into small parts, and providing immediate feedback. Along with the remarkable development of artificial intelligence, especially generative AI models like ChatGPT and Gemini, the potential to combine these two methods to create an optimized programming learning environment is becoming clear. Modern AI tools are capable of providing customized guidance and adaptive support, which are core features of PL, but with superior scale and flexibility. Furthermore, generative AI

opens up the possibility not only of supporting the learning process but also of revolutionizing the design and customization of programmed lessons, thereby strongly promoting self-learning and lifelong learning, contributing to the sustainable development of society.

This article focuses on analyzing the feasibility of integrating AI to support programming learning according to the PL method. We will delve into the principles of PL, the role of AI in programming education, and how AI can enhance the effectiveness of PL, especially in building personalized lessons and supporting lifelong learning. At the same time, an initial experiment in the form of a workshop on implementing Python programming learning with AI support at Mahasarakham University will be presented to illustrate the feasibility and potential of this approach.

2. BACKGROUND AND ANALYSIS

2.1. The Programmed Learning Method

The Programmed Learning method is a systematic pedagogical approach, developed based on B.F. Skinner's theory of operant conditioning in the mid-20th century (Skinner, 1954). The main goal of PL is to personalize the learning process, allowing learners to progress at their own pace and ability. PL operates on several basic principles (Hošková-Mayerová & Rosická, 2012; Mandić, 2012; Thomas et al., 1963):

- Breaking down knowledge (Small Steps): The learning content is divided into small "frames" of knowledge, each frame containing only a small amount of information and a corresponding question or exercise. This helps learners to absorb easily and minimizes the possibility of making errors.

- Immediate feedback: After each answer or completion of an exercise, the learner immediately receives information about the correctness of their answer. This feedback can be a confirmation (when correct) or corrective guidance (when incorrect). Immediate feedback is a key factor in reinforcing knowledge and adjusting learning behavior.

- Personalizing learning pace (Self-Pacing): Learners can decide their own learning speed, not being constrained by the general pace of the class. This ensures that each individual has enough time to deeply understand concepts before moving on to new content (Tran, D. Q. H., 2019).

- Active responding: Learners must actively participate in the learning process by providing answers or performing tasks instead of just passively receiving information.

- Logical sequencing: The knowledge frames are arranged in a strict logical sequence, from simple to complex, ensuring that learners build a solid foundation before moving on to more advanced concepts.

There are two main models of PL:

- Linear Programming: Proposed by Skinner, learners follow a fixed path through the frames. When they answer correctly, they move to the next frame. If they are wrong, they are prompted to try again or return to the previous frame. This model is presented in Figure 1 (Ha, V. H., 2014). In it, the knowledge frames are presented in

rectangles, the test parts in rhombuses, the arrow labeled "N" corresponds to the case of a wrong answer, and "Y" in the case of a correct answer.

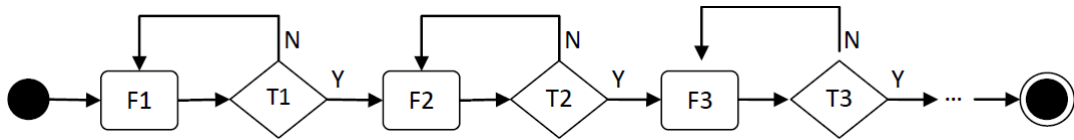


Figure 1. A Linear Programmed Lesson

- Branching Programming: Developed by Norman Crowder, it provides different learning paths depending on the learner's answer. If the answer is wrong, the learner will be directed to "remedial" or "supplementary" frames to re-explain the problem or provide more information before returning to the main content. This model allows for greater flexibility and personalization. Figure 2 (Ha, V. H., 2014) represents this model, with notations and conventions similar to the model in Figure 1.

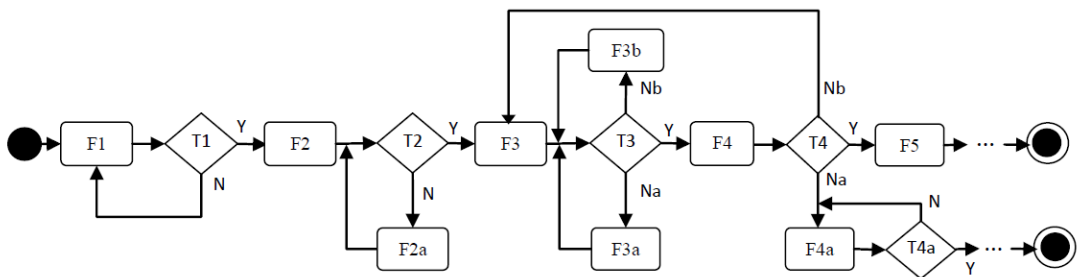


Figure 2. A Branching Programmed Lesson

PL has demonstrated benefits in improving learning outcomes and student interest, especially in subjects that require precision and logic such as mathematics and science (Darmayanti, 2024; Sadykov et al., 2023). For programming, a field that requires high precision and the ability to debug continuously, the principles of PL are very suitable for learners to gradually acquire knowledge and practice coding skills, thereby building a solid knowledge foundation and accurate practical ability.

2.2. Difficulties of the Traditional Programmed Learning Method

Although the Programmed Learning method offers many advantages in terms of personalization and immediate feedback, its implementation in practice, especially without the support of advanced technology like AI, has always faced significant challenges:

- Creating highly personalized programmed lessons requires enormous effort:

+ Compilation and design effort: The process of breaking down knowledge into logical frames, designing test questions for each frame, and especially building branching paths (in branching PL) to respond to different errors or learning needs of learners is an extremely time-consuming and laborious process. It requires the author not only to have deep expertise but also to have strong pedagogical skills to anticipate learning situations and potential errors.

+ Difficulty in personalizing at a large scale: Although PL aims at personalization, creating different versions of lessons or adjusting content for each specific learner with

diverse background knowledge, pace, and learning style is almost impossible if done manually. And no matter how much effort is put into detailed branching and personalization, overall, the programmed lesson is still a fixed entity, with a structure and content based on the subjective judgment of the designer, which cannot be customized for the learner at the time of use. This limits the scalability of the method, making its application to a large class or in the context of mass online learning impractical.

- Support for learners throughout the learning process is limited:

- + Static feedback and navigation: Traditional programmed lessons, whether presented in static text form or as computer programs, must have a predetermined path, designed in advance by the lesson creator. Although there is immediate feedback, this feedback is not flexible enough to resolve all questions or errors that fall outside the pre-built script. When a learner encounters an unexpected error or a concept they still cannot understand after several attempts, the traditional system can hardly provide dynamic, customized support.

- + Lack of real-time adaptability: Each individual's learning process is a constantly changing process. The learner's knowledge, skills, and interest level develop over time. The traditional PL method can hardly adapt in time to these changes, lacking the ability to adjust the difficulty, provide new examples, or change the explanation style flexibly throughout the learning process.

These difficulties have hindered the widespread adoption of the programmed learning method, especially in complex fields like programming, where the learning content changes frequently and learners often face problems that require flexible, continuous support.

2.3. Artificial Intelligence in supporting programming learning

Artificial Intelligence, especially large language models (LLMs) like ChatGPT and Gemini, is revolutionizing the way we approach learning. In programming education, AI has great potential to overcome the limitations of traditional teaching methods (Hartley et al., 2024; Le et al., 2013).

- Personalized and adaptive learning support: AI tools can provide "comprehensive and customized guidance on programming concepts and practices" (Hartley et al., 2024). They have the ability to synthesize information from multiple sources and present it in the form of integrated explanations with examples, suitable for the specific needs of each learner. This allows learners to progress at their own pace and receive support appropriate to their level of understanding.

- Intelligent Tutoring Systems (ITS): AI is widely used in ITS to support programming learning. Most existing systems use a "feedback-based tutoring approach" (Le et al., 2013). AI techniques in these systems can "identify the student's intention," that is, the solution strategy that the student has implemented in their solution. This allows AI to provide more accurate feedback and targeted guidance.

- Error analysis and correction suggestions: AI can analyze students' programming code, identify syntax errors, logical errors, and even suggest effective ways to fix them. This simulates the process of interacting with an experienced instructor.

- Content and exercise generation: Generative AI can create new programming exercises, sample code snippets, and practical scenarios, helping learners have a variety of resources for practice.

- Learning plan support: AI tools, such as ChatGPT, have been shown to be able to support planning by creating "detailed schedules" for learning (Hartley et al., 2024), helping learners organize their self-study process more effectively.

2.4. The combination of AI and the Programmed Learning method: overcoming traditional difficulties

The combination of AI and the Programmed Learning method creates a powerful learning system, leveraging the advantages of both, especially in overcoming the inherent difficulties of traditional PL. The concept of "Intelligent Educational Software" can "successfully replace or supplement teachers" in cases where the teacher's ability is technically or biologically limited (Mandic et al., 2012). This is especially true for PL, where AI can automate and optimize the core principles of this method, turning barriers into advantages:

- Automation and customization of lesson building, solving the difficulty of compilation effort and personalization:

+ Generative AI has the ability to strongly automate the process of creating knowledge frames, test questions, and even complex branching logic. This helps to significantly reduce the burden of manual compilation time and effort, which is the biggest barrier of traditional PL (Ha, V. H., 2014).

+ Based on detailed input requirements such as the learner's background knowledge and experience, the desired duration of the learning path, or the field the learner is studying/working in, AI can deeply customize the content. AI not only creates examples and exercises directly related to the learner's area of interest, but also automatically adjusts the difficulty, provides explanations suitable for their level, and even creates unique learning paths for each individual. This allows for "high personalization" on a large scale, which traditional PL could not achieve.

- Immediate, continuous, and large-scale personalized feedback, overcoming the limitation of learner support:

+ AI has the ability to provide immediate and accurate feedback to thousands of learners at the same time, something a teacher can hardly do. AI can analyze students' programming code, identify syntax errors, logical errors, and even suggest effective ways to fix them, far exceeding the capabilities of static feedback systems.

+ More importantly, AI can act as a personal "tutor," always ready to support the learner throughout the process. When a learner encounters difficulties or does not understand a concept, AI can provide more detailed explanations, different examples, or suggest new directions in a flexible and adaptive way in real time. This solves the problem of the lack of dynamic and adaptive support that traditional PL faced. AI can continuously adjust to the learner's progress and needs, providing a dynamic and always-accompanied learning environment.

- Enhancing autonomy and self-regulated learning (SRL) skills: AI promotes SRL by providing comprehensive support tools: instructional materials, content tools (practice environments, virtual compilers), automatic assessment, and learning planning support (Hartley et al., 2024).

- Adapting to learner diversity: AI allows learners to choose from multiple alternative answers, and if they choose incorrectly, the system will guide them to "re-investigate the entire previous work process and find where they made a mistake" (Mandic et al., 2012), which helps to reinforce a deeper understanding.

This integration not only helps to improve learning efficiency but also frees teachers from repetitive tasks such as basic grading, allowing them to focus on more complex educational activities such as developing critical thinking, problem-solving, and providing emotional support to learners.

3. WORKSHOP "AI-POWERED PROGRAMMING PEDAGOGY" AT MAHASARAKHAM UNIVERSITY

3.1. Context and objectives of the workshop

To explore the practical potential of applying generative AI to programming education according to the programmed method, a one-day workshop was organized for a group of graduate students from the Faculty of Education, Mahasarakham University (Thailand). This workshop did not aim to collect quantitative data for a full study, but was a preliminary survey to:

- Introduce the basic concepts of Python programming to those with no programming experience.

- Experience how generative AI (specifically through ChatGPT) can support the self-learning process according to the principles of programmed learning (i.e., providing information, feedback, and step-by-step guidance).

- Collect initial feedback and subjective opinions from participants about their experience with AI support.

3.2. Workshop design and implementation

The workshop was designed in the following stages:

- Introduction: A brief introduction to Python, the support capabilities of AI, specifically ChatGPT, in supporting programming learning/self-learning.

- AI-guided practice: Students were guided to use ChatGPT to self-learn basic Python programming. The main activities included:

+ How to generate a basic Python course for a period of 1 week.

+ How to follow the lessons in the above roadmap on a daily basis.

+ How to use AI support in each lesson and for each specific case.

▪ How to follow the daily schedule correctly.

▪ Reviewing the learned content.

- Generating a specific lesson for the day.
- Generating practice exercises, assessment questions after each learning frame;
- Seeking support when encountering problems: not understanding a theoretical content; wanting to learn more about a theoretical issue; having difficulty with a practical exercise or assessment question, etc.

+ To enhance thinking ability and deep understanding and proactively solve problems, students were guided to use AI as a tutor, avoiding dependence on AI and not letting AI completely replace the learner's thinking.

- Discussion and Feedback: At the end of the workshop, an open discussion was held for students to share their experiences and opinions on the effectiveness, strengths, and limitations of using AI in the process of self-learning programming.

3.3. Preliminary observations and feedback from the workshop

Following the workshop, all participants were invited to provide online feedback through a structured Google Form. Within one week, nine of the fifteen participants completed all items in the questionnaire. Although the workshop was relatively short and not designed to generate extensive quantitative data, the initial feedback yielded several valuable insights into participants' experiences.

The workshop consisted of fifteen students, all majoring in Science and Mathematics Education. Most participants had prior exposure to programming at the university level, with experience ranging from basic to intermediate proficiency. Two students had never undertaken any programming courses, while four had previously completed introductory Python programming classes. Interestingly, although many students reported regularly using AI tools in their general academic work, a majority of respondents (five out of nine) had never employed AI to support programming learning before participating in the workshop. Their feedback revealed several notable observations concerning students' capacity to integrate AI effectively into programming learning and self-directed study:

- Regarding the ability to generate learning content, from lesson roadmaps to specific content on theory, exercises, and assessment questions: AI can handle this completely, quickly, and with personalization based on input data, especially with the two data points on general programming experience and previous Python experience. Furthermore, the ability to customize and generate content during the learning process itself raises personalization to a new level.

- Regarding personalization in learning pace and content: Although observed only for the limited time of the workshop, this characteristic was clearly revealed. By the end of the workshop, students had reached almost different content points in the initially established learning path.

- The ability to use AI as a "tutor": This ability was established and trained right in the workshop, as a main objective to be achieved. Initially, when students encountered a problem, they often sought help from the student next to them, or asked the teacher. With the teacher's suggestions and guidance, through general principles as

well as specific methods and request commands (prompts), students gradually shifted to using AI's help to solve problems.

Through the workshop, two important results regarding the role of AI in sustainable development were drawn:

- On the potential to promote self-learning: The workshop suggested the idea that AI can be a powerful tool to encourage students to self-explore and self-learn programming, a foundation for lifelong learning in a sustainably developed society. 3/9 students planned to continue self-learning Python after the workshop, and most felt "quite confident" or "confident" in using AI to support programming self-learning in particular and self-learning in general.

- The role of AI as a tool for developing thinking: An important highlight in the feedback was the perception of how to use AI. One student emphasized: "AI should be used as a thinking assistant, not a replacement worker, and basic programming knowledge is still necessary for you to be able to control and supervise the results that AI generates." This shows an understanding of the need to combine AI with problem-solving skills and independent thinking. This is a key factor, helping people to exploit the strengths of AI without becoming dependent on it, thereby avoiding being replaced by AI in a sustainably developed society.

4. DISCUSSION: Towards lifelong learning and sustainable development

The analyses, experiences, and observations above can once again affirm that the learning model combining the Programmed Learning method and Generative Artificial Intelligence (AI-PL) not only brings direct benefits to the teaching and learning of programming, but also has broader impacts, contributing to the major goals of lifelong learning and sustainable development.

4.1. Contribution to the Sustainable Development Goals (SDGs)

In the global context, this approach directly dialogues with the United Nations' Sustainable Development Goal 4 (SDG 4), which emphasizes "ensuring inclusive and equitable quality education and promoting lifelong learning opportunities for all" (United Nations, 2015). International organizations such as UNESCO and the World Economic Forum (WEF) have identified AI as a key tool to realize this goal (UNESCO, 2021). The AI-PL model can be seen as a concrete example of responsibly applying AI to promote SDG 4 through two main mechanisms:

- Promoting equitable education through large-scale personalization: One of the biggest barriers to equitable education is the "one-size-fits-all" model. The traditional education system can hardly meet the diverse needs and learning paces of each individual. AI, as implemented in the AI-PL model, has the ability to provide a deeply customized learning experience for thousands of learners at the same time. This helps to narrow the gap between fast and slow learners, support those in difficult circumstances, and thereby enhance equity in education.

- Enhancing the role of the educator: WEF and UNESCO predict that the world will face a shortage of tens of millions of teachers to achieve SDG 4 by 2030. AI is not intended to replace teachers, but to "augment" their role. By automating repetitive

tasks such as creating exercises, basic grading, and answering frequently asked questions, the AI-PL model frees up time for teachers so they can focus on higher-value tasks such as mentoring, inspiring, and providing deep emotional and cognitive support to students.

4.2. Cultivating lifelong learning competency

In a world where the labor market and technology are constantly changing, the core value of the AI-PL model lies not only in teaching a specific skill like programming, but also in equipping learners with a meta-skill: the ability to self-learn and work effectively with the help of AI. This is one of the fundamental competencies for lifelong learning in the 21st century (Fullan & Scott, 2014; James, D., Sadik, S., & Brown, P., 2023).

Once a learner has mastered the learning process according to the AI-PL model, they can apply the same process to self-learn any other new skill in the future. Recent experimental studies have also shown that the effective use of AI tools is positively correlated with the job performance and academic success of lifelong learners (Ahn, H. Y., 2024). Our model provides a structured and safe environment to practice this competency right from the university level, preparing students for a future where the ability to retrain oneself becomes a survival factor.

4.3. Challenges of Overreliance on AI and Ethical Considerations

Despite its potential, the implementation of the AI-PL model faces several significant challenges and ethical considerations:

- Overreliance on AI: One of the key risks is that learners may become excessively dependent on AI-generated solutions, which can impede the development of essential problem-solving and critical thinking skills. To mitigate this, the pedagogical design should emphasize AI as a tutor or thinking assistant - a tool to support learning - rather than as a replacement performer.

- Ethical concerns: Issues such as data privacy, algorithmic bias in content generation, and the reliability of AI-produced information must be carefully addressed to ensure a fair, transparent, and trustworthy learning environment.

- Digital competence requirements: Successful application of this model demands that learners possess adequate digital literacy and metacognitive skills to critically engage with AI systems. This highlights the importance of foundational training to prepare learners for effective and responsible interaction with AI tools.

5. CONCLUSION AND FUTURE DIRECTIONS

The article has analyzed the powerful potential of integrating AI, especially generative AI, to support programming learning according to the Programmed Learning method. This combination brings the ability to personalize learning on a large scale, provide immediate and detailed feedback, and promote self-regulation skills. In particular, generative AI can revolutionize the process of building programmed lessons by creating highly customized content for each learner, based on their individual needs, experience, and goals. This not only optimizes learning efficiency but is also the key to promoting self-learning and lifelong learning

capabilities, turning the training process into a continuous self-training process, making an important contribution to the sustainable development of society. Preliminary findings from the workshop at Mahasarakham University have suggested the potential of AI in organizing, providing feedback, and coordinating the learning process, offering superior flexibility compared to traditional programmed methods, while providing initial insights into the learner's experience.

However, to maximize the benefits and minimize the challenges, further research and development are needed. Research prospects include:

- Developing smarter AI models: It is necessary to develop AI with the ability to understand the learning context more deeply, identify more complex types of errors, and provide more sophisticated feedback, while improving interaction and assessment capabilities.

- Designing integrated pedagogical models: Further research on how to design optimal curricula and learning processes, where AI and the programmed method are harmoniously integrated, not only supporting knowledge but also developing programming thinking and creative problem-solving skills.

- Evaluating long-term and quantitative effectiveness: There is a need for larger-scale and longer-term experimental studies to quantitatively evaluate the effectiveness of learning programming with AI support, including the impact on knowledge retention, problem-solving skills, and learner confidence, as well as the ability to maintain and develop lifelong learning skills.

- Building capacity for learners and teachers: Developing training programs and guidance materials to enhance learners' metacognitive skills in using AI, as well as equipping teachers with the necessary tools and knowledge to implement effectively.

The continuous development of AI promises to open new horizons for programming learning, turning it into a more personalized, engaging, and effective learning process, while equipping learners with the necessary tools to be autonomous and continuously learn throughout their lives, contributing to the sustainable development of society.

REFERENCES

- Ahn, H. Y. (2024). AI-powered e-learning for lifelong learners: Impact on performance and knowledge application. *Sustainability*, 16(20), 9066.
- Darmayanti, R. (2024). Programmed learning in mathematics education before and after the pandemic. *Assyfa Learning Journal*, 2(1), 40-56.
- Feldhusen, J. F., & Birt, A. (1962). A study of nine methods of presentation of programmed learning material. *The Journal of Educational Research*, 55(9), 461-466.
- Fullan, M., & Scott, G. (2014). *Education PLUS: The world will be led by people you can count on, including you*. Collaborative Impact SPC.
- Hà, V. H. (2004). Từ phương pháp dạy học chương trình hóa đến phương pháp học chương trình hóa [From programmed teaching method to programmed learning method]. *Tap chí Khoa học, Đại học Huế* [Journal of Science, Hue University], (22), 35-44.

- Hà, V. H. (2014). Easily creating programmed lessons with Microsoft PowerPoint and Adobe Presenter. *Tạp chí Khoa học và Giáo dục, Trường Đại học Sư phạm, Đại học Huế* [Journal of Science and Education, Hue University of Education], (4).
- Hartley, K., Hayak, M., & Ko, U. H. (2024). Artificial intelligence supporting independent student learning: An evaluative case study of ChatGPT and learning to code. *Education Sciences, 14*(2), 120.
- Hošková-Mayerová, Š., & Rosická, Z. (2012). Programmed learning. *Procedia - Social and Behavioral Sciences, 31*, 782-787.
- James, D., Sadik, S., & Brown, P. (2023). Rethinking Lifelong Learning in the “Fourth Industrial Revolution”. In *Third international handbook of lifelong learning* (pp. 1091-1110). Springer International Publishing.
- Le, N. T., Strickroth, S., Gross, S., & Pinkwart, N. (2013). A review of AI-supported tutoring approaches for learning programming. In *Proceeding of ICCSAMA 2013*. Springer.
- Mandic, D. (2012). Artificial intelligence in programmed teaching and learning. In *Proceedings of the 11th Wseas International Conference on Artificial Intelligence Knowledge Engineering and Data Bases*.
- Sadykov, T., Kokibasova, G., Minayeva, Y., Ospanova, A., & Kasymova, M. (2023). A systematic review of programmed learning approach in science education. *Cogent Education, 10*(1), 2189889.
- Skinner, B. F. (1954). The science of learning and the art of teaching. *Harvard Educational Review, 24*(2).
- Thomas, C. A., Davies, I. K., Openshaw, D., & Bird, J. B. (1963). *Programmed learning in perspective: A guide to program writing*. Routledge.
- Trần, D. Q. H. (2019). Nghiên cứu thiết kế phần mềm hỗ trợ ôn tập phân số (Toán 4) theo phương pháp chương trình hóa [A study on designing software to support reviewing fractions (Grade 4 Mathematics) using the programmed method]. *Tạp chí Khoa học Trường Đại học Sư phạm TP Hồ Chí Minh* [HCMUE Journal of Science], 59, 97.
- United Nations. (2015). *Transforming our world: the 2030 Agenda for Sustainable Development*.
- UNESCO. (2021). *AI and education: Guidance for policy-makers*.

TRÍ TUỆ NHÂN TẠO TẠO SINH VÀ PHƯƠNG PHÁP HỌC CHƯƠNG TRÌNH HÓA: NÂNG CAO NĂNG LỰC TỰ HỌC LẬP TRÌNH HƯỚNG TỚI HỌC TẬP SUỐT ĐỜI VÀ PHÁT TRIỂN BỀN VỮNG

Tóm tắt: Bài viết này phân tích tiềm năng của việc tích hợp Trí tuệ nhân tạo Tạo sinh (Generative AI) với phương pháp Học tập Chương trình hóa (Programmed Learning - PL) nhằm tạo ra một môi trường tự học lập trình được cá nhân hóa và hiệu quả. Dựa trên các lý thuyết về PL và Học tập Tự điều chỉnh (SRL), bài báo đề xuất một mô hình AI-PL mang tính khái niệm. Một đợt khảo sát ban đầu đã được tiến hành với các học viên cao học trong một workshop tại Đại học Mahasarakham (Thái Lan) để điều tra tính khả thi của việc sử dụng AI (ChatGPT) cho việc tự học Python. Phản hồi sơ bộ cho thấy AI có thể thực hiện hiệu quả vai trò tổ chức, cung cấp phản hồi thích ứng và điều phối quá trình học tập như hình dung của phương pháp PL. Hướng tiếp

cận này cho thấy tiềm năng trong việc cá nhân hóa nhịp độ, nội dung học tập và bồi dưỡng các kỹ năng tự học. Mô hình AI-PL được đề xuất có tiềm năng khắc phục những hạn chế truyền thống của PL, thúc đẩy năng lực học tập suốt đời và đóng góp vào Mục tiêu Phát triển Bền vững 4 (SDG4) bằng cách làm cho giáo dục lập trình trở nên dễ tiếp cận và công bằng hơn.

Từ khóa: AI, trí tuệ nhân tạo tạo sinh, học chương trình hóa, tự học lập trình, ChatGPT, học tập tự điều chỉnh, cá nhân hóa học tập, học tập suốt đời, phát triển bền vững.

INTELLIGENCE ARTIFICIELLE GÉNÉRATIVE ET MÉTHODE D'APPRENTISSAGE PROGRAMMÉ: RENFORCER LA CAPACITÉ D'AUTO-APPRENTISSAGE EN PROGRAMMATION EN VUE DE L'APPRENTISSAGE TOUT AU LONG DE LA VIE ET DU DÉVELOPPEMENT DURABLE

Résumé: Cet article analyse le potentiel de l'intégration de l'intelligence artificielle générative avec la méthode d'apprentissage programmé (PL) afin de créer un environnement d'auto-apprentissage personnalisé et efficace pour la programmation. S'appuyant sur les théories de l'apprentissage programmé et de l'apprentissage autorégulé (SRL), l'article propose un modèle conceptuel AI-PL. Un atelier exploratoire initial a été mené auprès d'étudiants de deuxième cycle à l'Université de Maharakham (Thaïlande) afin d'examiner la faisabilité de l'utilisation de l'IA (ChatGPT) pour l'auto-apprentissage de Python. Les retours préliminaires indiquent que l'IA peut remplir efficacement les rôles d'organisation, de fourniture de rétroactions adaptatives et de coordination du processus d'apprentissage, conformément aux principes de la méthode PL. Cette approche montre un potentiel pour personnaliser le rythme et le contenu d'apprentissage, tout en favorisant les compétences d'auto-apprentissage. Le modèle AI-PL proposé pourrait surmonter les limites traditionnelles de la PL, promouvoir les compétences d'apprentissage tout au long de la vie et contribuer à l'Objectif de Développement Durable 4 (ODD 4) en rendant l'enseignement de la programmation plus accessible et plus équitable.

Mots-clés: IA, intelligence artificielle générative, apprentissage programmé, auto-apprentissage de la programmation, ChatGPT, apprentissage autorégulé, apprentissage personnalisé, apprentissage tout au long de la vie, développement durable.